IN3015/4015 SONAR Imaging

September 2021

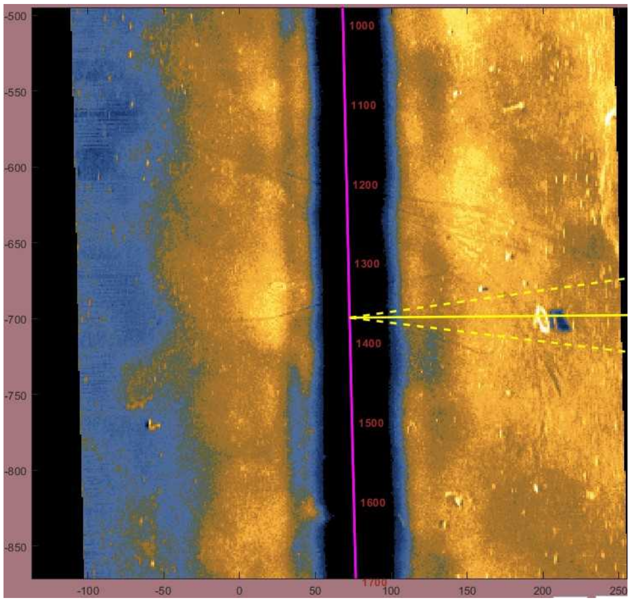

Figure 1: Sidescan sonar image of the scene

Introduction

The exercises in this project will be related to sonar data collected by the HUGIN autonomous underwater vehicle. Figure 1 shows the recorded sonar data stacked in one line per ping and shown in a two-dimensional image as function of range (x-axis) and ping number (y-axis). This is called a sidescan sonar image.

We will consider a single ping (or pulse) with recorded timeseries from one horisontal receiver array with $N_h = 32$ hydrophones (receivers). The transducer that emits the pulse can be approximated to a point source situated in the middle of the receiver array.

The data file `sonar_ping.uff` at the course web page, is a channel data object stored in a ultrasound file format file that contains the recorded timeseries from a single ping of sonar data as described above.

The length of the receiver array is $L = N_h d$, and the transmitter is assumed to be centered at the receiver array. You can find details about the probe in the `uff.probe` object found at `channel_data.probe` as well as other relevant parameters in the `channel_data`. You should explore the `channel_data` object in MATLAB.

The transmitted signal is a Linear Frequency Modulated (LFM) upchirp pulse of pulse length T_p, with signal bandwidth of B, as follows

$$s_{Tx}(t) = \begin{cases} \exp(j2\pi\alpha t^2/2) & -T_p/2 \le t \le T_p/2 \\ 0 & |t| > T_p/2 \end{cases} \tag{1}$$

Description	Variable	Units
Raw data	rawdata	
Sampling frequency	fs	Hz
Center frequency	fc	Hz
Bandwidth	bw	Hz
Pulse length	t_p	s
Element size	d	m
Sound speed	c	m/s

Table 1: Variables in the mat-file

where α is the chirp rate related to the signal bandwidth as

$$\alpha = B/T_p \tag{2}$$

Note that the received timeseries are basebanded at reception (the carrier frequency has been removed). We have therefore taken out the carrier (center frequency) from the signal model. Thus, the data is store as `IQ-data` as defined in lecture for module 3.

a)

- What is the theoretical angular resolution of the system at the center frequency?
- What is the angular field of view of the system at the center frequency?
- What is the range (depth) resolution?

All answers must contain both the expression and the numeric value for this particular system.

Pulse compression

Pulse compression can be performed either by cross correlating the received signal with the transmitted

$$s_m(\tau) = \int s_{Rx}(t) s_{Tx}^*(t+\tau) dt \tag{3}$$

or equivalently, using the Fourier transform

$$s_m(\tau) = \mathcal{FT}^{-1} \left\{ \mathcal{FT}\{s_{Rx}(t)\} \mathcal{FT}\{s_{Tx}(t)\}^* \right\}. \tag{4}$$

b)

See Fig. 2 a to verify your results in this task.

- Implement a pulse compression algorithm in MATLAB.
- Use Eq. (1) to generate a synthetic replica of the transmitted signal. Code must be presented.
- Pulse compress the rawdata. Choose a single channel, plot the magnitude of the particular channel before and after pulse compression. Plot the data zoomed in at sample number 4000 to 4400.
  - (!) Remember to only use positive delays if you use the MATLAB-function `xcorr`.
  - (!) Remember to complex conjugate if you use the Fourier transform based version.
- Describe the difference between the raw data and the pulse compressed data. Explain why there is a difference.

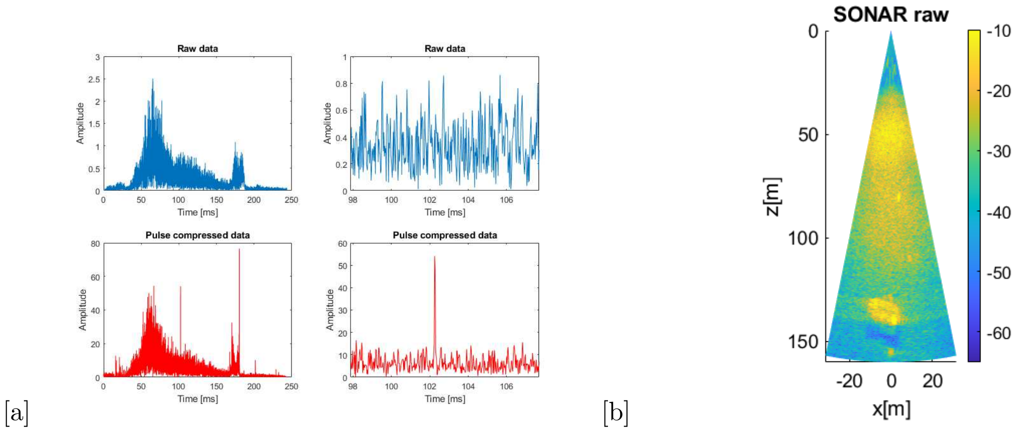

Figure 2: Example results from **b)** in [a] and **c)** in [b].

Beamforming with the USTB

You will beamform the sonar image with the UltraSound ToolBox (USTB) from both the raw channel data, and the data you have pulse compressed.

A good tip is to look at your previous modules on how beamforming with the USTB was set up in those exercises.

c)

- **Define the image scan** You will start of by defining a sector scan where you choose an output grid that matches the field of view of the system and the maximum range. Choose a grid resolution better than 20 cm in each dimension.
- **Parameters for the midprocess.das object** For this example you can simply use no receive and transmit wave apodization by using a `uff.window.none` for both the `transmit_apodization` and `receive_apodization`.
- **Run the processing object on both channel data objects** Run the delay-and-sum algorithm on the raw data and on the pulse compressed data.
- **Display** Display the sonar data before and after pulse compression, and before and after beamforming (four images) using relative intensity in dB-scale in cartesian coordinates. All plots should have meters on the range axis. The beamformed images should preferably have meters on both axis. You can use the built in plotting in the USTB to display the beamformed data.
- What is the difference before and after beamforming? Explain.
- What is the difference before and after pulse compression? Explain.
- Consider the pulse compressed beamformed (sectorscan sonar) image. There is a target at approximately 130 m range. What is it? How large is it in meters?